Fault Database for Southern California

July 17, 2003

Introduction - Team

Andrea Donnellan:
Principle Investigator,
Database design and implementation
Jet Propulsion Laboratory
Mail Stop 183-335
4800 Oak Grove Drive
Pasadena, CA 91109-8099
donnellan@jpl.nasa.gov
818-354-4737

Dennis McLeod:
Database interoperability
University of Southern California
Mail Code 0781
3651 Trousdale Parkway
Los Angeles, CA 90089-0742
mcleod@pollux.usc.edu
213-740-7285

Lisa Grant:
Fault database architect
University of California, Irvine
Environmental Analysis and Design
Irvine, CA 92697-7070
lgrant@uci.edu
949-824-5491

Academic Training:

Anne Yun-An Chen: University of Southern California, Ph.D. graduate student is undertaking development of the fault and federated databases.

Miryha M. Gould: University of California, Irvine, Ph.D. graduate student is developing the geological aspects of the fault database.

System Architecture

● Background

This database system manages a variety of types of earthquake science data and information. There are pre-existing collections, with heterogeneous access interfaces; there are also some structured collections managed by general-purpose database management systems. This new database enables the characterization of dynamically-defined earthquake faults.

In past work we have developed XML Document Type Definitions to describe various parameters of earthquake faults and input data. We developed this earthquake fault database based on this previous work. We continue to work with communities that have begun to establish data standards, such as the seismic community (effort led by Berkeley), and the International GPS Service. There has long been a need for establishing a database of faults for seismic hazard analysis (MG95).

Several databases have been constructed for this purpose by the U.S. Geological Survey (USGS), California Division of Mines and Geology (CDMG) and the Southern California Earthquake Center (SCEC), each with a different format. The primary goal of the existing databases, and current collaborative efforts by USGS, CDMG and SCEC on the "RELM" database, is to provide input for probabilistic assessment of ground motion parameters. Existing databases are not compatible and are not suitably formatted or readily accessible for simulations. For example, much of the focus has been on establishing whether or not certain faults exist or are "active" as defined by the state of California, and how their proposed geometries would affect ground motion estimates.

Most faults in the existing databases have been divided into characteristic segments that are proposed to rupture as a unit. Geologic slip rates are assigned to large segments rather than to the specific locations (i.e. geographic coordinates) where they were measured. These simplifications and assumptions are desirable for seismic hazard analysis, but they introduce a level of geologic interpretation and subjective bias that is inappropriate for simulations of fault behavior. This database includes primary geologic and paleoseismic fault parameters (fault location/geometry, slip rate at measured location, measurements of coseismic displacement, dates and locations of previous ruptures,) as well as separate interpreted/subjective fault parameters [GL99] such as characteristic segments, average recurrence interval, magnitude of characteristic ruptures etc. Both parameter types will be updated as more data is acquired and interpreted through research and the numerical simulations.

To support this earthquake fault database and others, we acquired and employed a state-of-the-art commercially-available general-purpose database management system (MySQL). In particular, we utilized a basic relational database management system running on a PC under LINUX. These systems support the definition, storage, access, and control of collections of structured data. To meet the needs of the simulation community, we aim to support the following features:

- extensible type definition capabilities in the database management system (to accommodate application-specific kinds of data) – **already developed in a prototype form**,
- the ability to combine information from multiple databases – **in the design, not yet developed**, and
- mechanisms to efficiently return XML results from requests – **near-term future work**.

● Design Requirements

Our system allows the user to operate on the data through an Internet connection currently at http://infogroup.usc.edu:8080/index.html . Prospective users should contact one of the authors above for user name and password. The user interface is browser-based. Java Script has to be supported at the user's side. The requirements of our system are the following:

1. Hardware Requirements (HR):
   HR1: The user shall have a computer that has network access
   HR2: The server shall have internet connection
   HR3: The server shall allow Hypertext Transfer Protocol (HTTP)
2. Software Requirement (SR)
   SR1: The user shall have web browsers
   SR2: The PC on the user's side shall support Java Script program
   SR3: The server shall have an operating system that supports multi-tasking
   SR4: The server shall have one database management system for keeping data
   SR5: The server shall deal with the web base request from users
3. Security Requirement (ER)
   ER1: The system shall only allow Hypertext Transfer Protocol (HTTP) for the end user
4. Scalability Requirements (AR)
   AR1: The system administrator may preprogram the system anytime.
5. System Feedback Requirements (FR)
   FR1: The system shall pop-up an error message dialog box after invalid data input
   FR2: The system shall provide enough information about request

processing status.

6. <u>Performance Requirement (PR)</u>
   PR1: The system shall provide a response time within 10 seconds.

7. <u>Operational Requirements (OR)</u>
   OR1: The system shall provide a method of inserting data
   OR2: The system shall provide a method of deleting data
   OR3: The system shall provide a method of reviewing data
   OR4: The system shall provide the contact information for the end user
   OR5: The system shall provide a log of all events generated during operation.

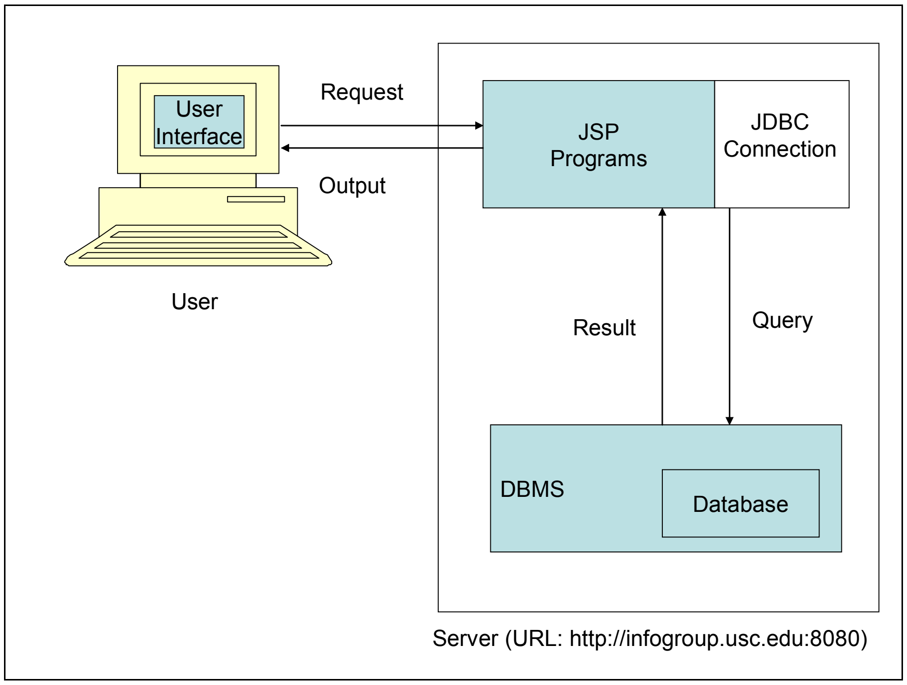


Fig 1. System Architecture

- Operating System
  The operating system installed in our server is Linux, Red Hat 8.0.

- Web Server
  We installed Apache and Jakarta Tomcat V. 4.1 on our server in order to support web pages and JavaServer Pages (JSP).

- Database Management System
  We use MySQL as our backend database management system. MySQL is easy to get and operate.

Implementation

- User Interface

We use hypertext markup language (HTML) and JavaScript to program our user interfaces. HTML is used on the World Wide Web and has provided a good base for web pages for a long time. We employ JavaScript to check input errors and show reminders to the end users. Our user interfaces are in the formats of forms to provide simple but sufficient functions.

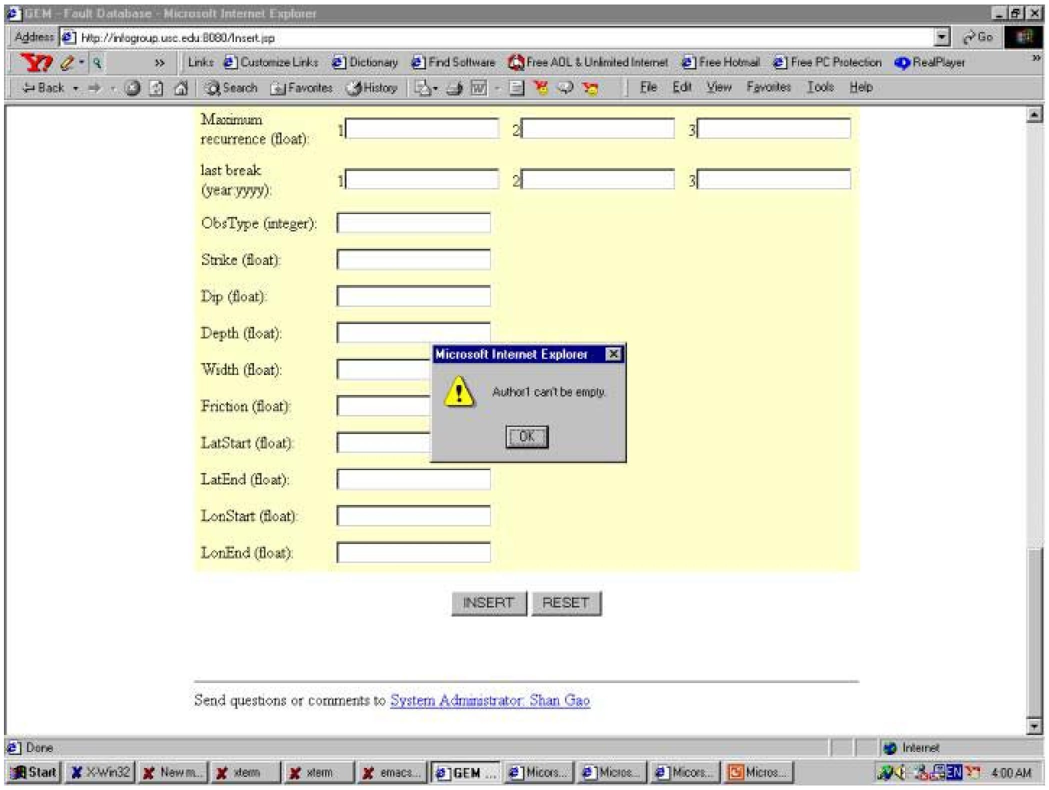
When the user clicks on submit (here means “insert”, “select”, or “delete”), the backend corresponding JSP programs will be called to further process the request.

Functionality

- Insert Data

Insert data function is handled by two programs: Insert.jsp and insert.jsp.

Insert.jsp:
Has four purposes: getting parameters, examining the validation of data, showing confirmation message, and passing parameters to insert.jsp. The validation of data falls into two categories: existence and format. Existence means that certain parameters must be assigned some values. In Fig 2, we show one example of our error message regarding this.


GEM - Fault Database - Microsoft Internet Explorer
Address http://infogroup.usc.edu:8080/Insert.jsp
Back
Search
Favorites
History
File Edit View Favorites Tools Help
INSERT
RESET
OK
Done
Internet
Start
4:00 AM

Maximum recurrence (float):	1	2	3
last break (year:yyyy):	1	2	3
ObsType (integer):			
Strike (float):			
Dip (float):			
Depth (float):			
Width (float):			
Friction (float):			
LatStart (float):			
LatEnd (float):			
LonStart (float):			
LonEnd (float):			

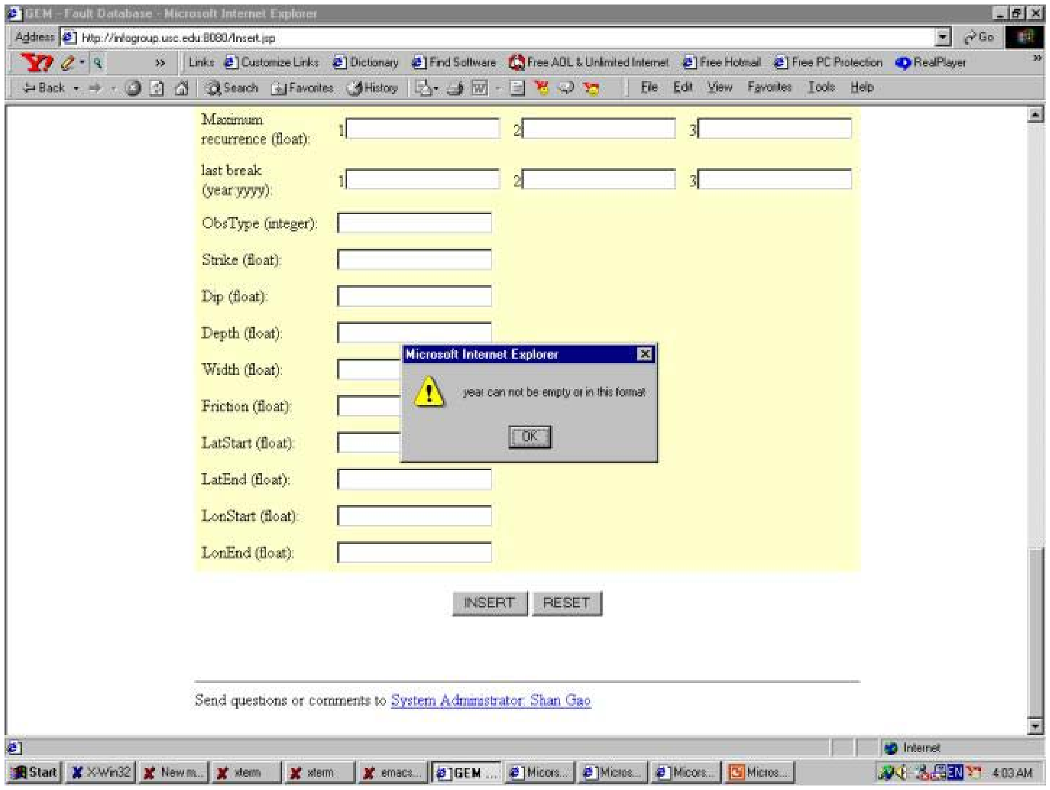
Microsoft Internet Explorer

Author1 can't be empty.

Send questions or comments to System Administrator: Shan Gao

Fig 2. Error message for requiring value for parameter Author1

Some parameters have to be entered in specified formats. For example, parameter Year has to be a four-digit and an integer number. In Fig 3, we display another example of our error message.


GEM - Fault Database - Microsoft Internet Explorer
Address http://infogroup.usc.edu:8080/Insert.jsp
Links Customize Links Dictionary Find Software Free AOL & Unlimited Internet Free Hotmail Free PC Protection RealPlayer
Back Search Favorites History File Edit View Favorites Tools Help

Maximum recurrence (float):	1	2	3
last break (year:yyyy):	1	2	3
ObsType (integer):			
Strike (float):			
Dip (float):			
Depth (float):			
Width (float):			
Friction (float):			
LatStart (float):			
LatEnd (float):			
LonStart (float):			
LonEnd (float):			

Microsoft Internet Explorer

year can not be empty or in this format

OK

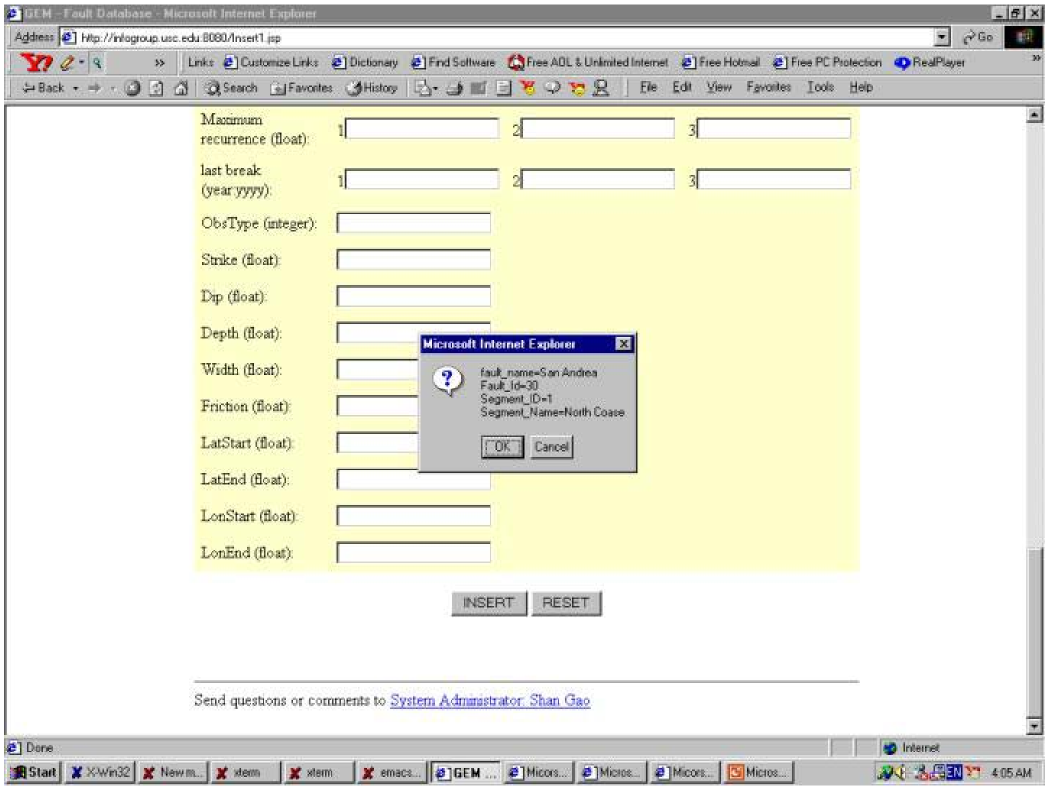
INSERT RESET

Send questions or comments to System Administrator: Shan Gao

Internet
Start X-Win32 New m... xterm xterm emacs... GEM ... Micros... Micros... Micros... Micros... 4:03 AM

Fig 3. Error message for incorrect format

A confirmation message will pop-up before the parameters passed to insert.jsp. Values of four parameters, Fault ID, Fault Name, Segment ID, and Segment Name, will be showed in the pop-up window and the system will ask our user to confirm the inputs. In Fig 4, we show the confirmation message.



GEM - Fault Database - Microsoft Internet Explorer

Address: http://infogroup.usc.edu:8080/Insert1.jsp

Maximum recurrence (float):	1	2	3
last break (year:yyyy):	1	2	3
ObsType (integer):			
Strike (float):			
Dip (float):			
Depth (float):			
Width (float):			
Friction (float):			
LatStart (float):			
LatEnd (float):			
LonStart (float):			
LonEnd (float):			

Microsoft Internet Explorer

```
fault_name=San Andrea
Fault_Id=30
Segment_ID=1
Segment_Name=North Coase
```

Send questions or comments to System Administrator: Shan Gao

Fig 4. Confirmation message

insert.jsp

insert.jsp will get the parameters from Insert.jsp, form SQL queries, call JDBC connection to our DBMS, and execute queries. After successfully executing queries, a message will be shown in the browser to inform the user the completion of insertion.

- Select Data

Select data function is handled by two programs: Select.jsp and select.jsp.

Select.jsp:

Select.jsp performs three actions: getting parameters, examining the validation of data, and passing parameters to select.jsp. These actions are the same as those of Insert.jsp.

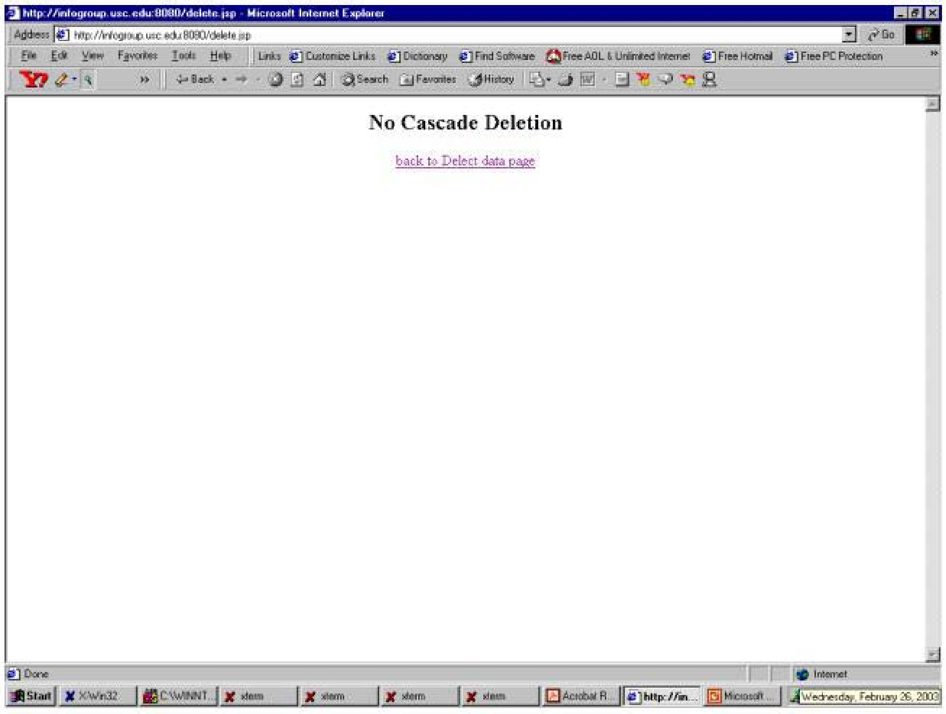
select.jsp
select.jsp will get the parameters from Select.jsp, form SQL queries, call JDBC connection to our DBMS, execute queries, and print out the query results in the browser. If no parameter is specified, all data in Fault Database would be shown.

- Delete Data

Delete data function is handled by two programs: Delete.jsp and delete.jsp.

Delete.jsp:
This program will get parameters, examine the validation of data, and pass parameters to delete.jsp. These actions are the same as those of Insert.jsp.

delete.jsp
It will get the parameters from Delete.jsp, form SQL queries, call JDBC connection to our DBMS, execute queries, and print out the query results in the browser. We don't allow cascade deletion. Therefore, an error message will be shown if the user doesn't specify any parameter. In Fig 5, we display the error message regarding to this.



No Cascade Deletion

back to Delect data page

Fig 5. Error message regarding to forbidden cascade deletion

- Insert Layer Data

Insert layer data function is handled by two programs: Insert_Layer.jsp and insert_Layer.jsp.

Insert_Layer.jsp:
Like Insert.jsp, this page is used for four purposes: getting parameters, examining the validation of data, showing confirmation message, and passing parameters to insert_Layer.jsp.

insert_Layer.jsp:
insert_Layer.jsp will get the parameters from Insert_Layer.jsp, form SQL queries, call JDBC connection to our DBMS, and execute queries. After successfully executing queries, a message will be shown in the browser to inform the user the completion of insertion.

- Select Layer Data

Select layer data function is handled by two programs: Select_Layer.jsp and select_Layer.jsp.

Select_Layer.jsp:
Select_Layer.jsp performs three actions: getting parameters, examining the validation of data, and passing parameters to select_Layer.jsp. These actions are the same as those of insert_Layer.jsp.

select_Layer.jsp
select_Layer.jsp will get the parameters from Select_Layer.jsp, form SQL queries, call JDBC connection to our DBMS, execute queries, and print out the query results in the browser. If no parameter is specified, all data in Fault Database for Layer would be shown.

Data in Fault Database

- Source
  Data in the fault database are extracted from refereed journal articles, professional papers, professional reports, and conference abstracts.

- Parameters
  The following parameters are the particular properties that describe the data in Fault Database:

Parameter	**Full Name**	**Definition**
Fault_id	Fault ID number	Fault ID numbers are extracted from the Southern California Earthquake Center (SCEC) Phase II report (Working Group on California Earthquake Probabilities, 1995)
FaultName	Fault name	Fault names are taken from study site descriptions provided in each individual data source. Faults are fractures dividing two bodies of rock, along which the rock masses have moved against each other.

Parameter	Full Name	Definition
StrandName	Strand name	Strand names are taken from study site descriptions provided in each individual data source. Fault strands are fault branches or splays of a larger fault zone.
InterpId	Interpretation ID number	The interpretation ID numbers are automatically generated for each individual data source entered into the fault database
Author	Author names	The authors of each publication are listed in authorship order.
Publication	Publication title	The name of the journal article, professional paper, professional report, or conference abstract.
Year	Year	The year of publication of the source.
Title	Title	The name of the journal, professional paper, professional report, or conference proceedings which features the publication.
Volume	Volume	The volume number of the journal, professional paper, professional report, or conference proceedings which features the publication.
Number	Number	The number or issue of the journal, professional paper, professional report, or conference proceedings which features the publication.
Pages	Pages	The page numbers of the publication.
Comment	Comment	The comment field allows for additional explanation regarding the data source.
SegmentId	Segment ID number	Segment ID numbers are assigned for each fault segment using a rubrick designed by Miryha M. Gould, based on segments recognized by Petersen et al., 1996.
SegmentName	Segment name	Segment names are taken from study site descriptions provided in each individual data source. Only segments recognized by Petersen et al., 1996 are considered at this point.
Strike	Strike	The geographic orientation of a fault plane. Specifically, strike is the direction of a horizontal line in the plane of a fault.
Dip	Dip	The slope or vertical component of a fault plane.
Depth	Depth	The distance from the surface of the earth to the base of a fault or fault zone.

Parameter	Full Name	Definition
Width	Width	The surface or subsurface extent of a fault zone.
LatStart	Latitude start	The latitude of the location of one end of a fault. LatStart corresponds with LonStart.
LatEnd	Latitude end	The latitude of the location of one end of a fault. LatEnd corresponds with LonEnd.
LonStart	Longitude start	The longitude of the location of one end of a fault. LonStart corresponds with LatStart.
LonEnd	Longitude end	The longitude of the location of one end of a fault. LonEnd corresponds with LatEnd.
Datum	Datum	The name of the geographic datum corresponding to the latitude and longitude points describing the location of the fault at the surface of the Earth. Geographic datums define the size and shape of the earth and the orientation of the coordinate systems used to map the Earth.
LastBreak	Last break	The year of the last earthquake rupture along a section or at a specific location on a fault. Quantities are expressed in calendar years.
Friction	n/a	This is a future implementation.
ObsType	Observation type	The method of observation used by the source authors for data collection. For example, paleoseismology or geomorphology.
AveRecurr	Average recurrence interval	The mean time between earthquakes of a given magnitude or magnitude range on a fault. Quantities are expressed in years.
MinRecurr	Minimum recurrence interval	The minimum time between earthquakes of a given magnitude or magnitude range on a fault. Quantities are expressed in years.
MaxRecurr	Maximum recurrence interval	The maximum time between earthquakes of a given magnitude or magnitude range on a fault. Quantities are expressed in years.
AveSlip	Average slip per event	The mean amount of displacement on a fault averaged over the number of earthquake events observed. Quantities are expressed in meters.
MinSlip	Minimum slip per event	The minimum amount of displacement on a fault averaged over the number of earthquake events observed. Quantities are expressed in meters.

Parameter	Full Name	Definition
MaxSlip	Maximum slip per event	The maximum amount of displacement on a fault averaged over the number of earthquake events observed. Quantities are expressed in meters.
AveDipSlip	Average dip slip per event	The mean amount of displacement perpendicular to the strike of a fault averaged over the number of earthquake events observed. Quantities are expressed in meters.
MinDipSlip	Minimum dip slip per event	The minimum amount of displacement perpendicular to the strike of a fault averaged over the number of earthquake events observed. Quantities are expressed in meters.
MaxDipSlip	Maximum dip slip per event	The maximum amount of displacement perpendicular to the strike of a fault averaged over the number of earthquake events observed. Quantities are expressed in meters.
AveStrikeSlip	Average strike slip per event	The mean amount of horizontal displacement parallel to the strike of a fault averaged over the number of earthquake events observed. Quantities are expressed in meters.
MinStrikeSlip	Minimum strike slip per event	The minimum amount of horizontal displacement parallel to the strike of a fault averaged over the number of earthquake events observed. Quantities are expressed in meters.
MaxStrikeSlip	Maximum strike slip per event	The maximum amount of horizontal displacement parallel to the strike of a fault averaged over the number of earthquake events observed. Quantities are expressed in meters.
AveDipRate	Average dip slip rate	The mean rate of displacement perpendicular to the strike of a fault averaged over a time period representing one to several large earthquakes. Quantities are expressed in millimeters per year.
MinDipRate	Minimum dip slip rate	The minimum rate of displacement perpendicular to the strike of a fault averaged over a time period representing one to several large earthquakes. Quantities are expressed in millimeters per year.

Parameter	Full Name	Definition
MaxDipRate	Maximum dip slip rate	The maximum rate of displacement perpendicular to the strike of a fault averaged over a time period representing one to several large earthquakes. Quantities are expressed in millimeters per year.
AveStrikeRate	Average strike slip rate	The mean rate of horizontal displacement parallel to the strike of a fault averaged over a time period representing one to several large earthquakes. Quantities are expressed in millimeters per year.
MinStrikeRate	Minimum strike slip rate	The minimum rate of horizontal displacement parallel to the strike of a fault averaged over a time period representing one to several large earthquakes. Quantities are expressed in millimeters per year.
MaxStrikeRate	Maximum strike slip rate	The maximum rate of horizontal displacement parallel to the strike of a fault averaged over a time period representing one to several large earthquakes. Quantities are expressed in millimeters per year.

The other group, Material Rectilinear Layer, has the following parameters:

- Initialization:
  LayerName
  LayerID

- Geographic and geometric description:
  LatOrigin (analogous to LatEnd, LonEnd for a SEGMENT)
  LonOrigin
  Datum
  OriginX
  OriginY
  OriginZ (These three are in km, location of the block SW corner from the coordinate axis origin).
  Length
  Width
  Depth

- Material constants (numbers) with their units (strings):

  LameLambda

  LameLambdaUnits

  LameMu

  LameMuUnits

  Viscosity

  ViscosityUnits

  ViscosityExponent

Performance Evaluation

● Requirement Verification and Evaluation

Legend:

NC	– No Change from initial specification
Mo	– Modified from initial specification, see justification for description
Add	– The new requirement which was added after initial specification
NI	– Not Implemented in prototype, see justification for description
I	– Implemented in prototype

Hardware Requirements (HR)

Req. #	Status	Justification
HR1	NC, I	This is implemented.
HR2	NC, I	This is implemented. The domain name of the server is infogroup.usc.edu.
HR3	NC, I	This is implemented. The port for HTTP has been opened since finishing Apache server installation.

Software Requirement (SR)

Req. #	Status	Justification
SR1	NC, I	This is implemented.
SR2	NC, NI	Due to the variety of users, a message shall be shown to inform the user about the JavaScript support.
SR3	NC, I	This is implemented. Linux, Red Hat 8.0 has been installed on our server.
SR4	NC, I	This is implemented. MySQL, the DBMS, has been installed on our server.
SR5	NC, I	This is implemented. Jakarta Tomcat V. 4.1 has been installed and started up on our server in order to support web pages and JavaServer Pages (JSP)

Security Requirement (ER)

Req. #	Status	Justification
ER1	NC, I	This is implemented. We've closed all the ports except HTTP and SSH for administrative purpose.

Scalability Requirements (AR)

Req. #	Status	Justification
AR1	Mo, I	The requirement has been modified so that not only system administrator but also the affiliate for this project.

System Feedback Requirements (FR)

Req. #	Status	Justification
FR1	NC, I	This is implemented. JavaScript codes has been added to the web pages to check the validation of the input data.
FR2	NC, I	This is implemented. The query results would be shown to the user.

Performance Requirement (PR)

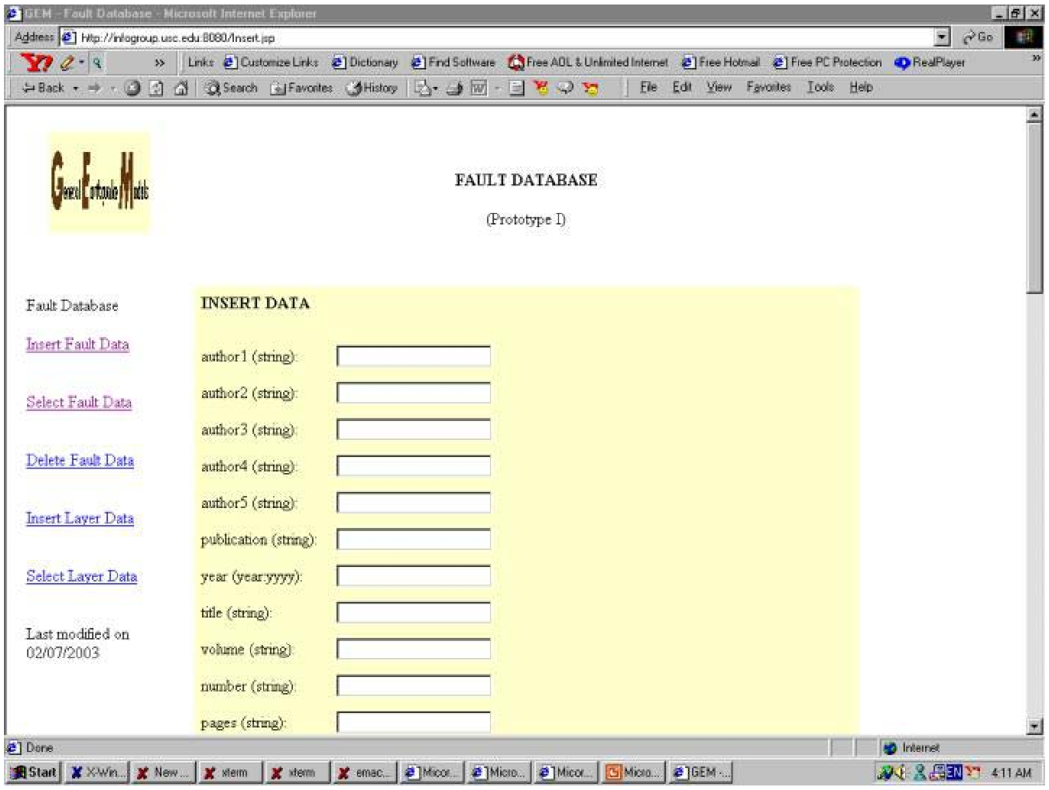
Req. #	Status	Justification
PR1	NC, I	This is implemented. We programmed simple web pages to display the results in a short time.

Operational Requirements (OR)

Req. #	Status	Justification
OR1	NC, I	This is implemented. We've coded for two programs, Insert.jsp and insert.jsp, to handle the functionalities of data insertions.
OR2	NC, I	This is implemented. We've coded for two programs, Select.jsp and select.jsp, to handle the functionalities of data selections.
OR3	NC, I	This is implemented. We've coded for two programs, Delete.jsp and delete.jsp, to handle the functionalities of data deletions.
OR4	NC, I	This is implemented. The user is able to click the link and generate mails to the system administrator.
OR5	NC, I	This is implemented. Jakarta Tomcat records every event in the log file.

- Web Pages for Successful Functionalities

- Insert Data



FAULT DATABASE

(Prototype I)

Fault Database

Insert Fault Data

Select Fault Data

Delete Fault Data

Insert Layer Data

Select Layer Data

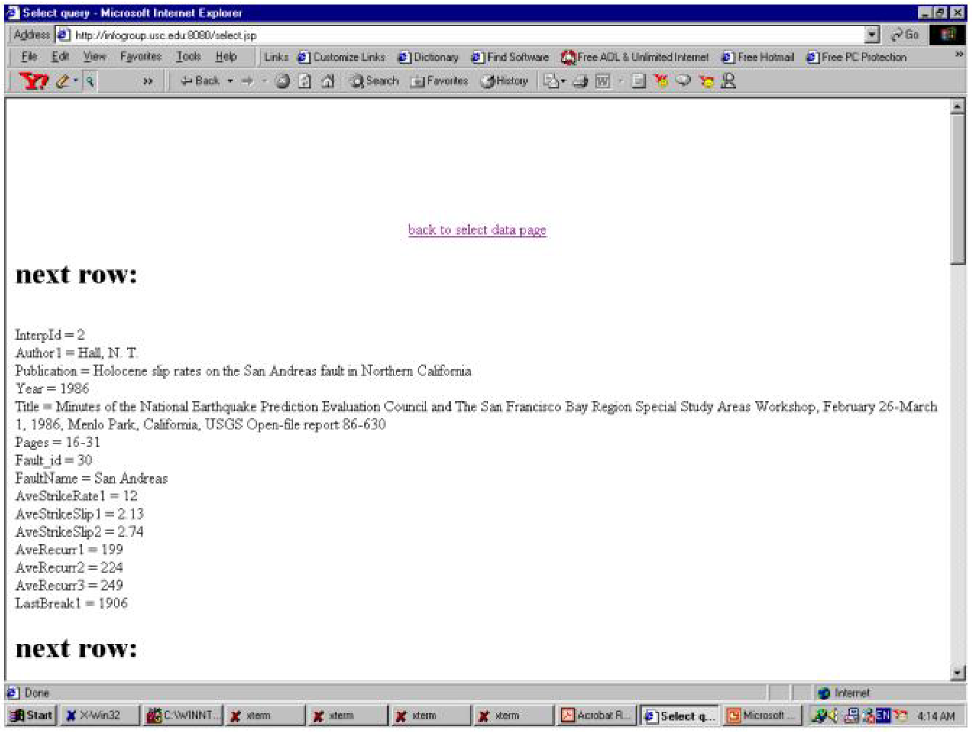
Last modified on
02/07/2003

INSERT DATA

author1 (string):	
author2 (string):	
author3 (string):	
author4 (string):	
author5 (string):	
publication (string):	
year (year:yyyy):	
title (string):	
volume (string):	
number (string):	
pages (string):	

Fig 6. Insert.jsp

- Select Data


Select query - Microsoft Internet Explorer
Address http://infogroup.usc.edu:8080/select.jsp
File Edit View Favorites Tools Help
Back Search Favorites History
Done
Internet

back to select data page

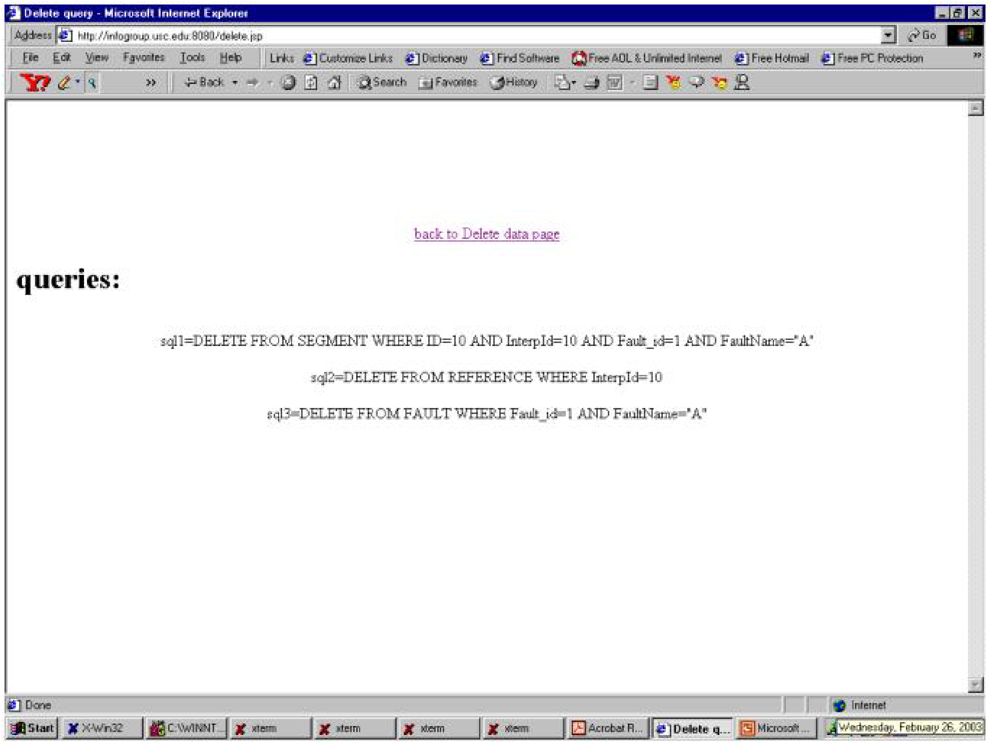
next row:

InterpId = 2
Author1 = Hall, N. T.
Publication = Holocene slip rates on the San Andreas fault in Northern California
Year = 1986
Title = Minutes of the National Earthquake Prediction Evaluation Council and The San Francisco Bay Region Special Study Areas Workshop, February 26-March 1, 1986, Menlo Park, California, USGS Open-file report 86-630
Pages = 16-31
Fault_id = 30
FaultName = San Andreas
AveStrikeRate1 = 12
AveStrikeSlip1 = 2.13
AveStrikeSlip2 = 2.74
AveRecurr1 = 199
AveRecurr2 = 224
AveRecurr3 = 249
LastBreak1 = 1906

next row:

Fig 7. Query result for selecting data

- Delete Data


Delete query - Microsoft Internet Explorer
http://infogroup.usc.edu:8080/delete.jsp
back to Delete data page

queries:

```
sql1=DELETE FROM SEGMENT WHERE ID=10 AND InterpId=10 AND Fault_id=1 AND FaultName="A"

sql2=DELETE FROM REFERENCE WHERE InterpId=10

sql3=DELETE FROM FAULT WHERE Fault_id=1 AND FaultName="A"
```

Fig 8. Showing the queries for deleting data

Future Work

The fault database has been designed to accommodate several types of fault data and data sets, including primary data, non-primary summary data, and simulated or hypothetical data. The fault database is currently populated with a small number of examples drawn from primary sources such as journal articles and conference abstracts. The examples focus on paleoseismic data from major faults in California. The current examples do not contain much information on geometric fault attributes, or geographic coordinates of fault segments because these are not typically published in research papers.

Future work will focus on populating the fault database with primary paleoseismic data, and with structured data sets containing summary fault attributes and geographic coordinates of fault segments. We will enter more paleoseismic data from research publications, and add two larger structured data sets: Virtual California, and the fault database published by the California

Geological Survey (CGS) for seismic hazard analysis. The CGS data set provides geographic coordinates, geometry, and summary attributes for many active faults and fault segments in California. We also intend to improve the effectiveness of database queries by enabling wildcard searches.

Appendix – Database Schema

In the following, we discuss the database schema with SQL statements. These statements have actually been used to create tables in the database.

SQL statements defining the schema for Fault Database:

```
create table FAULT
    (Fault_id int not null,
    FaultName char(255) not null,
    StrandName char(255),
    primary key(Fault_id, FaultName) );

create table REFERENCE
    (InterpId INT NOT NULL auto_increment,
    Author1 char(255),
    Author2 char(255),
    Author3 char(255),
    Author4 char(255),
    Author5 char(255),
    Publication char(255),
    Year int,
    Title char(255),
    Volume char(255),
    Number char(255),
    Pages char(255),
    Comment char(255),
    primary key(InterpId) );

create table SEGMENT
    (Fault_id int not null,
    FaultName char(255) not null,
    InterpId INT NOT NULL,
    SegmentId int,
    SegmentName char(255),
    Strike float,
    Dip float,
    Depth float,
    Width float,
    LatStart float,
    LatEnd float,
    LonStart float,
    LonEnd float,
    LastBreak1 float,
    LastBreak2 float,
    LastBreak3 float,
    Friction float,
    ObsType int,
    AveRecurr1 float,
    AveRecurr2 float,
    AveRecurr3 float,
    MinRecurr1 float,
    MinRecurr2 float,
    MinRecurr3 float,
    MaxRecurr1 float,
    MaxRecurr2 float,
    MaxRecurr3 float,
    AveSlip1 float,
    AveSlip2 float,
    AveSlip3 float,
    MinSlip1 float,
    MinSlip2 float,
    MinSlip3 float,
    MaxSlip1 float,
    MaxSlip2 float,
    MaxSlip3 float,
    AveDipSlip1 float,
    AveDipSlip2 float,
    AveDipSlip3 float,
    MinDipSlip1 float,
    MinDipSlip2 float,
    MinDipSlip3 float,
    MaxDipSlip1 float,
    MaxDipSlip2 float,
    MaxDipSlip3 float,
    AveStrikeSlip1 float,
    AveStrikeSlip2 float,
    AveStrikeSlip3 float,
    MinStrikeSlip1 float,
    MinStrikeSlip2 float,
    MinStrikeSlip3 float,
    MaxStrikeSlip1 float,
    MaxStrikeSlip2 float,
    MaxStrikeSlip3 float,
    AveDipRate1 float,
    AveDipRate2 float,
    AveDipRate3 float,
    MinDipRate1 float,
    MinDipRate2 float,
    MinDipRate3 float,
    MaxDipRate1 float,
    MaxDipRate2 float,
    MaxDipRate3 float,
    AveStrikeRate1 float,
    AveStrikeRate2 float,
    AveStrikeRate3 float,
    MinStrikeRate1 float,
    MinStrikeRate2 float,
    MinStrikeRate3 float,
    MaxStrikeRate1 float,
    MaxStrikeRate2 float,
    MaxStrikeRate3 float,
    Foreign key (Fault_id) references FAULT(Fault_id),
    Foreign key (FaultName) references FAULT(FaultName),
    Foreign key (InterpId) references REFERENCE(Interpid) );
```

create table LREFERENCE

```
(InterpId INT NOT NULL auto_increment,
Author1 char(255),
Author2 char(255),
Author3 char(255),
Author4 char(255),
Author5 char(255),
Publication char(255),
Year int,
Title char(255),
Volume char(255),
Number char(255),
Pages char(255),
Comment char(255),
primary key(InterpId) );
```

create table LAYER

```
(InterpId INT NOT NULL,
LayerId int not null,
LayerName char(255),
LatOrigin float,
LonOrigin float,
Datum char(255),
OriginX float,
OriginY float,
OriginZ float,
Length float,
Width float,
Depth float,
LameLambda float,
LameLambdaUnits char(255),
LameMu float,
LameMuUnits char(255),
Viscosity float,
ViscosityUnits char(255),
ViscosityExponent float,
primary key(InterpId) );
```